

## Práctica 1. Sistema de pago para un e-commerce



### → INTRODUCCIÓN

El profe turroneiro de *Lenguaje de Marcas* os ha contado tantas cosas sobre comercio online que os ha convencido: vais a crear un sitio web para estudiantes de programación que se dedique a vender cursos sobre tecnologías varias.

### → PROBLEMA A RESOLVER

Imagina que estás desarrollando el sistema de pagos para tu tienda de cursos online. La tienda acepta diferentes métodos de pago, como tarjeta de crédito, *PayPal* o *Bizum*. Cada método de pago tiene una forma distinta de procesar el pago, pero todos comparten la misma funcionalidad básica para hacerlo: un método llamado ***procesarPago()***.

Realiza un programa en *Java* que implemente la lógica de la aplicación explicada anteriormente, usando POO mediante herencia, polimorfismo y abstracción.

### Construcción de las clases que van a interactuar entre ellas

La aplicación que necesitamos desarrollar constará de los siguientes elementos:

- Una clase abstracta llamada ***MetodoPago***, con el siguiente método abstracto:

***void procesarPago(double importe)***

- Crea 3 subclases que extiendan y hereden de *MetodoPago*, una por cada tipo de método de pago permitido: *TarjetaCredito*, *PayPal* y *Bizum*.
- La clase ***TarjetaCredito*** tendrá los atributos *nro\_tarjeta* (*String* de 16 caracteres) y *tipo* (*String* que debe permitir solamente los siguientes valores: *VISA*, *MASTERCARD*, *MAESTRO*).
  - El método ***procesarPago()*** debe imprimir "*Procesando pago de [importe] € con tarjeta de crédito VISA*".

- Debe haber un método más en la clase para **validarTarjeta()**, que compruebe que el *nro\_tarjeta* introducido tenga la longitud esperada y el *tipo* esté dentro de los valores permitidos.
- La clase **PayPal** tendrá los atributos *cuenta* (*String* con formato de correo electrónico "xxx@xxx.com") y *saldo* (*double* por defecto 23€).
  - El método **procesarPago()** debe imprimir "Procesando pago de [importe] € con PayPal".
  - En este caso, el método **validarPayPal()** debe comprobar que el formato del correo electrónico es correcto y que el *saldo* de la cuenta sea suficiente para realizar el pago correspondiente.
- La clase **Bizum** tendrá los atributos *telefono* (*String* de 9 caracteres) y *pin* (*int* de 6 dígitos que se generará de forma aleatoria).
  - El método **procesarPago()** debe imprimir "Procesando pago de [importe] € con Bizum".
  - El método **validarBizum()** debe comprobar el formato del teléfono y que el pin introducido por el usuario es el correcto. **NOTA:** haz trampa e imprime el pin en cuanto se genere para poder ver cuál es y poder realizar el pago.
- Tendremos otra clase **Tienda** con un método estático:

***static void realizarPago(MetodoPago metodo)***

que pedirá el importe a pagar e invocará al método *procesarPago()* según el tipo de objeto *metodo* recibido como parámetro.

- Además, esta clase tendrá otro método estático **iniciarPago()**, donde previamente preguntará al usuario qué método de pago quiere usar para crear uno nuevo y realizar todas las validaciones correspondientes (*validarTarjeta*, *validarPayPal*, *validarBizum*) antes de ser procesado.
- Si lo necesitas, crea más métodos auxiliares para modularizar el código todo lo que puedas.
- Por último, crea un programa principal **AppEcommerce** que implemente toda la lógica.

```
public class AppEcommerce {  
  
    public static void main(String[] args) {  
  
        Tienda.iniciarPago();  
  
    }  
  
}
```

## Ejemplos de ejecución

Método incorrecto:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Pr
¿Qué metodo de pago quieres usar? [Bizum, Paypal, Tarjeta]
applepay
El método de pago no existe.

Process finished with exit code 0
|
```

Pago tarjeta ok:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Progra
¿Qué metodo de pago quieres usar? [Bizum, Paypal, Tarjeta]
tarjeta
Introduce los datos de tu tarjeta:
8574837485982345
Selecciona el tipo de tarjeta [VISA, MAESTRO, MASTERCARD]:
VISA
Validando tarjeta...
Introduce el importe a pagar:
20
Procesando pago de 20.0 € con tarjeta de crédito VISA
Pago aceptado. Muchas gracias.

Process finished with exit code 0
|
```

Tarjeta incorrecta:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Pr
¿Qué metodo de pago quieres usar? [Bizum, Paypal, Tarjeta]
tarjeta
Introduce los datos de tu tarjeta:
234
Selecciona el tipo de tarjeta [VISA, MAESTRO, MASTERCARD]:
VISA
Validando tarjeta...
Los datos de tu tarjeta no son correctos.

Process finished with exit code 0
|
```

Pago Bizum ok:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Pr
¿Qué metodo de pago quieres usar? [Bizum, Paypal, Tarjeta]
bizum
Introduce tu número de teléfono vinculado con Bizum:
6666666666
[chivato PIN] 235432
Introduce tu PIN:
235432
Validando Bizum...
Introduce el importe a pagar:
20
Procesando pago de 20.0 € con Bizum.
Pago aceptado. Muchas gracias.

Process finished with exit code 0
```

Bizum incorrecto:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Pr
¿Qué metodo de pago quieres usar? [Bizum, Paypal, Tarjeta]
bizum
Introduce tu número de teléfono vinculado con Bizum:
6666666666
[chivato PIN] 235432
Introduce tu PIN:
123456
Validando Bizum...
Los datos de tu Bizum no son correctos.

Process finished with exit code 0
```

## → REALIZACIÓN DE LA PRÁCTICA

Sigue los siguientes pasos para realizar la práctica. **¡Ve guardando tu trabajo de vez en cuando para evitar que se borre el avance si se cierra el editor de textos u ocurre cualquier problema en tu equipo!**

1. Programa en Java la aplicación requerida
2. Plan de pruebas:

- a. Realiza las pruebas manuales necesarias para comprobar que el programa funciona bien.
- b. Pruebas unitarias de los método implementados con JUnit5.

### 3. Comentarios de Javadoc

### 4. Diagrama UML con PlantUML y ChatGPT.



## ENTREGA

**REALIZA UN INFORME EN *FORMATO MARKDOWN* CON LA INFO GENERADA Y LOS PASOS SEGUIDOS PARA REALIZAR ESTA PRÁCTICA. EXPLICA TU CÓDIGO.**

**SUBE TODO A TU REPOSITORIO DE *GITHUB* Y PEGA LA URL EN LA TAREA DE AULES DISPONIBLE.**